UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/015,530	12/12/2001	Robert T. Plunkett	021202-000900US	7763

20350 7590 07/14/2004

TOWNSEND AND TOWNSEND AND CREW, LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER
O BRIEN, BARRY J

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 07/14/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/015,530	PLUNKETT ET AL.	
	Examiner	Art Unit	
	Barry J. O'Brien	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12/12/01, 5/3/02, 11/21/02, 6/21/04*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *03 May 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *6/21/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-25 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed on record in the file: Declaration and Fee as received on 5/3/02, Drawings as received on 5/3/02, Request to Rescind Previous Nonpublication Request as received on 11/21/02, and IDS as received on 6/21/04.

Specification

3. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

4. The applicant is requested to review the specification and update the status of all co-pending applications made mention of, replacing attorney docket numbers with current U.S. application or patent numbers when appropriate.

Double Patenting

5. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

6. Claims 1, 6, 11, 16 and 21 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 8, 15, 22 and 29, respectively, of copending Application No. 10/015,544. Although the conflicting claims are not identical, they are not patentably distinct from each other because claims 1, 6, 11, 16 and 21 of the instant application are obvious variations of claims 1, 8, 15, 22 and 29 of the copending application. For example, claim 11 of the instant application regard first and second groups of heterogeneous computational elements being reconfigurable to form first and second functional units to implement first and second functions, respectively, and further wherein if the second function is idle, one or more of the second group of heterogeneous computational elements are reconfigurable to implement the first function. Claim 15 of the copending application recites the same limitations as instant claim 11, except that instead of first and second functions, it recites a system acquisition function and a communication function. The first and second functions of instant claim 11 are obvious variations of the system acquisition and communication functions of copending claim 15. Claims 1, 6, 16 and 21 are also obvious variations of claims 1, 8, 22 and 29, respectively.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8. Claims 21-25 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claim 21 is directed towards software that is to perform configuring and reconfiguring on groups of heterogeneous computational elements. This is non-statutory, as there is no hardware structure claimed on which the software can be executed. It is suggested that claim 21 be amended to be directed towards a "computer-implemented method" in order to make the claim statutory. Dependent claims 22-25 are rejected for the same reasons as above, as they contain all of the limitations of the rejected parent claim.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

10. Claims 21 and 23-25 are rejected under 35 U.S.C. 102(b) as being anticipated by New, U.S. Patent No. 6,046,603, (hereinafter New(1)), including New et al., U.S. Patent No.6,091,263 (hereinafter New(2)), which is incorporated by reference (see New(1), Col.3 lines 50-61).

11. Regarding claim 21, New has taught a method for allocating hardware resources within an adaptive computing integrated circuit, comprising:

a. In response to first configuration information, configuring a first group of heterogeneous computational elements (see New(1), 9-16 of Fig.1 and Col.5 lines 39-46) to form a first functional unit to implement a first function (see New(2), Col.1 lines 13-20 and Col.3 lines 25-42) and configuring a second group of heterogeneous computational elements (see New(1), 1-8 of Fig.1 and Col.5 lines 39-46) to form a second functional unit to implement a second function (see New(2), Col.1 lines 13-20 and Col.3 lines 25-42). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

b. In response to second configuration information, reconfiguring one or more of the second group of heterogeneous computational elements to implement the first function (see New(1), Col.5 line 39 – Col.6 line 51). Because New does not include specific language excluding having the same function implemented more than once, and by teaching that any function can be implemented on the selected group of computational elements, New therefore allows the same function to be implemented multiple times as needed by the reconfiguration information.

12. Regarding claim 23, New has taught the method of claim 21, wherein in response to the second configuration information, the one or more of the second group of heterogeneous computational elements are reconfigured to form one or more additional instances of the first functional unit to implement the first function (see above paragraph 11). Here, the group of

computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires. Because New does not include specific language excluding having the same function or functional unit implemented more than once, and by teaching that any function or functional unit can be implemented on the selected group of computational elements, New therefore allows the same function or functional unit to be implemented multiple times as needed by the reconfiguration information.

13. Regarding claim 24, New has taught the method of claim 21, wherein in response to the second configuration information, one or more of the first group of heterogeneous computational elements and the one or more of the second group of heterogeneous computational elements are reconfigured to form a single functional unit to implement the first function (see above paragraph 11). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Because New does not include specific language excluding having the same function or functional unit implemented more than once, and by teaching that any function or functional unit can be implemented on the selected rectangular group of computational elements, New therefore allows the same function or functional unit to be implemented multiple times as needed by the reconfiguration information.

14. Regarding claim 25, New has taught the method of claim 21, further comprising:

a. In response to third configuration information, reconfiguring one or more of the first group of heterogeneous computational elements and/or the one or more of the second group of heterogeneous computational elements to implement a third function (see above paragraph 11). Here, any rectangular group of computational

elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

Claim Rejections - 35 USC § 103

15. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

16. Claims 1-20 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over New, U.S. Patent No. 6,046,603, (hereinafter New(1)), including New et al., U.S. Patent No.6,091,263 (hereinafter New(2)), which is incorporated by reference (see New(1), Col.3 lines 50-61), in further view of Wirthlin et al., *A Dynamic Instruction Set Computer* (hereinafter Wirthlin).

17. Regarding claim 1, New has taught an adaptive computing integrated circuit configurable to perform a plurality of functions, comprising:

a. A plurality of heterogeneous computational elements (see New(1), 1A-24A and 1B-24B of Fig.1),

b. An interconnection network coupled to the plurality of heterogeneous computational elements (see New(1), Fig.1), the interconnection network operative to configure the plurality of heterogeneous computational elements (see New(1), Col.5 line 39 – Col.6 line 51),

c. Wherein a first group of heterogeneous computational elements (see New(1), 9-16 of Fig.1 and Col.5 lines 39-46) is configurable to form a first functional unit to implement a first function (see New(2), Col.1 lines 13-20 and Col.3 lines 25-42),

d. Wherein a second group of heterogeneous computational elements (see New(1), 1-8 of Fig.1 and Col.5 lines 39-46) is configurable to form a second functional unit to implement a second function (see New(2), Col.1 lines 13-20 and Col.3 lines 25-42),

e. Wherein one or more of the second group of heterogeneous computational elements are reconfigurable by the interconnection network to implement the first function (see New(1), Col.5 line 39 – Col.6 line 51). Here, the group of computational elements can be partially reconfigured to implement any function the reconfiguration information requires. Because New does not include specific language excluding having the same function implemented more than once, and by teaching that any function can be implemented on the selected group of computational elements, New therefore allows the same function to be implemented multiple times as needed by the reconfiguration information.

18. New has not explicitly taught wherein if the second function is not currently used, one or more of the second group of heterogeneous computational elements are reconfigurable by the interconnection network to implement the first function.

19. However, Wirthlin has taught the reconfiguring of groups of computational elements (instruction modules) so as to replace idle instruction modules at run-time, thus improving the partial re-configuration speed and overall throughput of the system (see Wirthlin, Sec. 2.2).

Because New has taught a method of partial reconfiguration of groups of computational elements (see New(1), Col.5 lines 39-46), one of ordinary skill in the art would have found it obvious to modify the adaptive computing integrated circuit of New to reconfigure idle groups of computational elements in order to reduce the partial reconfiguration speed and increase overall throughput.

20. Regarding claim 2, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 1, wherein if the second function is not currently used, the one or more of the second group of heterogeneous computational elements are reconfigurable to implement the first function by forming one or more additional instances of the first functional unit (see above paragraphs 17-19). Here, the group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires. Because New does not include specific language excluding having the same function or functional unit implemented more than once, and by teaching that any function or functional unit can be implemented on the selected group of computational elements, New therefore allows the same function or functional unit to be implemented multiple times as needed by the reconfiguration information.

21. Regarding claim 3, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 1, wherein if the second function is not currently used, one or more of the first group of heterogeneous computational elements and the one or more of the second group of heterogeneous computational elements are reconfigurable to form a single functional unit to implement the first function (see above paragraphs 17-19). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any

functional unit the reconfiguration information requires (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Because New does not include specific language excluding having the same function or functional unit implemented more than once, and by teaching that any function or functional unit can be implemented on the selected rectangular group of computational elements, New therefore allows the same function or functional unit to be implemented multiple times as needed by the reconfiguration information.

22. Regarding claim 4, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 1, wherein if the second function is not currently used, the one or more of the second group of heterogeneous computational elements are reconfigurable by the interconnection network to implement one or more of the plurality of functions other than the second function (see above paragraphs 17-19). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

23. Regarding claim 5, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 1, wherein if a third function is to be implemented, one or more of the first group of heterogeneous computational elements and/or the one or more of the second group of heterogeneous computational elements are reconfigurable by the interconnection network to implement the third function (see above paragraphs 17-19). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

24. Regarding claim 6, New has taught an adaptive integrated circuit, comprising:

a. A plurality of reconfigurable matrices (see New(2), 601L, 601R of Fig.8 and Col.6 lines 5-10), the plurality of reconfigurable matrices including a plurality of heterogeneous computational units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46), each heterogeneous computational unit having a plurality of fixed computational elements (see New(1), 1A-24A and 1B-24B of Fig.1), the plurality of fixed computational elements including a first computational element having a first architecture (see New(1), 1A-24A of Fig.1) and a second computational element having a second architecture (see New(1), 1B-24B of Fig.1), the first architecture distinct from the second architecture (see New(1), Col.3 lines 41-60), the plurality of heterogeneous computational units coupled to an interconnect network (see New(1), Fig.1) and reconfigurable in response to configuration information (see New(1), Col.5 line 39 – Col.6 line 51). Here, any rectangular group of CLB's is considered a reconfigurable computational unit, and are thus heterogeneous as they don't have to be the same size and shape (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

b. A matrix interconnection network coupled to the plurality of reconfigurable matrices (see New(2), Fig.6, Fig.7 and Col.6 lines 28-57), the matrix interconnection network operative to reconfigure the plurality of reconfigurable matrices in response to the configuration information for a plurality of operating modes (see New(2), Col.6 lines 32-38),

c. Wherein a first group of heterogeneous computational units is reconfigurable to form a first functional unit to implement a first operating mode (see New(1),

Col.1 lines 56-60 and Col.5 lines 39-46). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

d. Wherein a second group of heterogeneous computational units is reconfigurable to form a second functional unit to implement a second operating mode (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

e. Wherein one or more of the second group of heterogeneous computational units are reconfigurable to implement the first operating mode (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Because New does not include specific language excluding having the same function implemented more than once, and by teaching that any function can be implemented on the selected group of

computational elements, New therefore allows the same function to be implemented multiple times as needed by the reconfiguration information.

25. New has not explicitly taught wherein if the second operating mode is not currently used, one or more of the second group of heterogeneous computational units are reconfigurable to implement the first operating mode.

26. However, Wirthlin has taught the reconfiguring of groups of computational units (instruction modules) so as to replace idle computational units at run-time, thus improving the partial re-configuration speed and overall throughput of the system (see Wirthlin, Sec. 2.2). Because New has taught a method of partial reconfiguration of groups of computational units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46), one of ordinary skill in the art would have found it obvious to modify the adaptive computing integrated circuit of New to reconfigure idle groups of computational units in order to reduce the partial reconfiguration speed and increase overall throughput.

27. Regarding claim 7, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 6, wherein if the second operating mode is not currently used, the one or more of the second group of heterogeneous computational units are reconfigurable to implement the first operating mode by forming one or more additional instances of the first functional unit (see above paragraphs 24-26). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Because New does not include specific language excluding having the same function implemented more than once, and

by teaching that any function can be implemented on the selected group of computational elements, New therefore allows the same function to be implemented multiple times as needed by the reconfiguration information.

28. Regarding claim 8, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 6, wherein if the second operating mode is not currently used, one or more of the first group of heterogeneous computational units and the one or more of the second group of heterogeneous computational units are reconfigurable to form a single functional unit to implement the first operating mode (see above paragraphs 24-26). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Because New does not include specific language excluding having the same function implemented more than once, and by teaching that any function can be implemented on the selected group of computational elements, New therefore allows the same function to be implemented multiple times as needed by the reconfiguration information.

29. Regarding claim 9, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 6, wherein if the second operating mode is not currently used, the one or more of the second group of heterogeneous computational units are reconfigurable to implement one or more of the plurality of operating modes other than the second operating mode (see above paragraphs 24-26). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration

information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

30. Regarding claim 10, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 6, wherein if a third operating mode is to be implemented, one or more of the first group of heterogeneous computational units and/or the one or more of the second group of heterogeneous computational units are reconfigurable to implement the third operating mode (see above paragraphs 24-26). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

31. Regarding claim 11, New has taught an adaptive computing integrated circuit, comprising:

a. A plurality of heterogeneous computational elements (see New(1), 1A-24A and 1B-24B of Fig.1), the plurality of heterogeneous computational elements including a first computational element and a second computational element, the first computational element having a first fixed architecture of a plurality of fixed architecture (see New(1), 1A-24A of Fig.1) and the second computational element having a second fixed architecture of the plurality of fixed architectures (see New(1), 1B-24B of Fig.1), the first fixed architecture being different than the second fixed architecture (see New(1), Col.3 lines 41-60), and the plurality of fixed architectures including functions for memory, addition, multiplication, complex multiplication, subtraction, configuration, reconfiguration, control, input,

output, and field programmability (see New(1), Col.3 lines 41-60). Here, the first computational elements (1A-24A of Fig.1) provide control functions to enable/disable reconfiguration circuitry, and the second computational elements (1B-24B of Fig.1) provide configuration and reconfiguration functions to the circuit.

b. An interconnection network coupled to the plurality of heterogeneous computational elements (see New(1), Fig.1), the interconnection network operative to configure the plurality of heterogeneous computational elements (see New(1), Col.5 line 39 – Col.6 line 51),

c. Wherein a first group of heterogeneous computational elements (see New(1), 9-16 of Fig.1 and Col.5 lines 39-46) is reconfigurable to form a first functional unit to implement a first function (see New(2), Col.1 lines 13-20 and Col.3 lines 25-42),

d. Wherein a second group of heterogeneous computational elements (see New(1), 1-8 of Fig.1 and Col.5 lines 39-46) is reconfigurable to form a second functional unit to implement a second function (see New(2), Col.1 liens 13-20 and Col.3 liens 25-42),

e. Wherein one or more of the second group of heterogeneous computational elements are reconfigurable by the interconnection network to implement the first function (see New(1), Col.5 line 39 – Col.6 line 51). Here, the group of computational elements can be partially reconfigured to implement any function the reconfiguration information requires. Because New does not include specific language excluding having the same function implemented more than once, and

by teaching that any function can be implemented on the selected group of computational elements, New therefore allows the same function to be implemented multiple times as needed by the reconfiguration information.

32. New has not explicitly taught wherein if the second function is not currently used, one or more of the second group of heterogeneous computational elements are reconfigurable by the interconnection network to implement the first function.

33. However, Wirthlin has taught the reconfiguring of groups of computational elements (instruction modules) so as to replace idle instruction modules at run-time, thus improving the partial re-configuration speed and overall throughput of the system (see Wirthlin, Sec. 2.2). Because New has taught a method of partial reconfiguration of groups of computational elements (see New(1), Col.5 lines 39-46), one of ordinary skill in the art would have found it obvious to modify the adaptive computing integrated circuit of New to reconfigure idle groups of computational elements in order to reduce the partial reconfiguration speed and increase overall throughput.

34. Regarding claim 12, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 11, wherein if the second function is not currently used, the one or more of the second group of heterogeneous computational elements are reconfigurable to implement the first function by forming one or more additional instances of the first functional unit (see above paragraphs 31-33). Here, the group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires. Because New does not include specific language excluding having the same function or functional unit implemented more than once, and by teaching that any function

or functional unit can be implemented on the selected group of computational elements, New therefore allows the same function or functional unit to be implemented multiple times as needed by the reconfiguration information.

35. Regarding claim 13, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 11, wherein if the second function is not currently used, one or more of the first group of heterogeneous computational elements and the one or more of the second group of heterogeneous computational elements are reconfigurable to form a single functional unit to implement the first function (see above paragraphs 31-33). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Because New does not include specific language excluding having the same function or functional unit implemented more than once, and by teaching that any function or functional unit can be implemented on the selected rectangular group of computational elements, New therefore allows the same function or functional unit to be implemented multiple times as needed by the reconfiguration information.

36. Regarding claim 14, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 11, wherein if the second function is not currently used, the one or more of the second group of heterogeneous computational elements are reconfigurable by the interconnection network to implement one or more of the plurality of functions other than the second function (see above paragraphs 31-33). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

37. Regarding claim 15, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 11, wherein if a third function is to be implemented, one or more of the first group of heterogeneous computational elements and/or the one or more of the second group of heterogeneous computational elements are reconfigurable by the interconnection network to implement the third function (see above paragraphs 31-33). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

38. Regarding claim 16, New has taught an adaptive computing integrated circuit, comprising:

a. A plurality of heterogeneous computational elements (see New(1), 1A-24A and 1B-24B of Fig.1), the plurality of heterogeneous computational elements including a first computational element and a second computational element, the first computational element having a first fixed architecture (see New(1), 1A-24A of Fig.1) and the second computational element having a second fixed architecture (see New(1), 1B-24B of Fig.1), the first fixed architecture being different than the second fixed architecture (see New(1), Col.31 lines 41-60),

b. An interconnection network coupled to the plurality of heterogeneous computational elements (see New(1), Fig.1), the interconnection network operative to configure a first group of heterogeneous computational elements to form a first functional unit for a first functional mode of a plurality of functional modes (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46 and New(2), Col.1

lines 13-20 and Col.3 lines 25-42), in response to first configuration information (see New(1), Col.5 line 39 – Col.6 line 51), and the interconnection network further operative to reconfigure a second group of heterogeneous computational elements to form a second functional unit for a second functional mode of the plurality of functional modes (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46 and New(2), Col.1 lines 13-20 and Col.3 lines 25-42), in response to second configuration information (see New(1), Col.5 line 39 – Col.6 line 51), the first functional mode being different than the second functional mode (see New(2), Col.1 lines 13-20 and Col.3 lines 25-42), and the plurality of functional modes including linear algorithmic operations, non-linear algorithmic operations, finite state machine operations (see New(2), Col.3 line 66 – Col.4 line 65), memory operations, and bit-level manipulations (see New(2), Col.4 line 66 – Col.5 line 11). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

c. Wherein one or more of the second group of heterogeneous computational units are reconfigurable to implement the first functional mode (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of

computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Because New does not include specific language excluding having the same function implemented more than once, and by teaching that any function can be implemented on the selected group of computational elements, New therefore allows the same function to be implemented multiple times as needed by the reconfiguration information.

39. New has not explicitly taught wherein if the second functional mode is not currently used, one or more of the second group of heterogeneous computational units are reconfigurable to implement the first functional mode.

40. However, Wirthlin has taught the reconfiguring of groups of computational units (instruction modules) so as to replace idle computational units at run-time, thus improving the partial re-configuration speed and overall throughput of the system (see Wirthlin, Sec. 2.2). Because New has taught a method of partial reconfiguration of groups of computational units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46), one of ordinary skill in the art would have found it obvious to modify the adaptive computing integrated circuit of New to reconfigure idle groups of computational units in order to reduce the partial reconfiguration speed and increase overall throughput.

41. Regarding claim 17, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 16, wherein if the second functional mode is not currently used, the one or more of the second group of heterogeneous computational elements are reconfigurable to implement the first functional mode by forming one or more additional instances of the first functional unit (see above paragraphs 38-40). Here, any rectangular group of computational

elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Because New does not include specific language excluding having the same function implemented more than once, and by teaching that any function can be implemented on the selected group of computational elements, New therefore allows the same function to be implemented multiple times as needed by the reconfiguration information.

42. Regarding claim 18, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 16, wherein if the second functional mode is not currently used, one or more of the first group of heterogeneous computational elements and the one or more of the second group of heterogeneous computational elements are reconfigurable to form a single functional unit to implement the first functional mode (see above paragraphs 38-40). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46). Because New does not include specific language excluding having the same function implemented more than once, and by teaching that any function can be implemented on the selected group of computational elements, New therefore allows the same function to be implemented multiple times as needed by the reconfiguration information.

43. Regarding claim 19, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 16, wherein if the second functional mode is not currently used, the one or more of the second group of heterogeneous computational elements are reconfigurable by

the interconnection network to implement one or more of the plurality of functional modes other than the second functional mode (see above paragraphs 38-40). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

44. Regarding claim 20, New in view of Wirthlin has taught the adaptive computing integrated circuit of claim 16, wherein if a third functional mode is to be implemented, one or more of the first group of heterogeneous computational elements and/or the one or more of the second group of heterogeneous computational elements are reconfigurable by the interconnection network to implement the third functional mode (see above paragraphs 38-40). Here, any rectangular group of computational elements can be partially reconfigured to implement any function or act as any functional unit the reconfiguration information requires, including groups of computational elements that are grouped together as functional units (see New(1), Col.1 lines 56-60 and Col.5 lines 39-46).

45. Regarding claim 22, New has taught the method of claim 21, but has not explicitly taught wherein the second configuration information is generated when the second function is not currently used.

46. However, Wirthlin has taught the reconfiguring of groups of computational elements (instruction modules) so as to replace idle instruction modules at run-time, thus improving the

partial re-configuration speed and overall throughput of the system (see Wirthlin, Sec. 2.2). Because New has taught a method of partial reconfiguration of groups of computational elements (see New(1), Col.5 lines 39-46), one of ordinary skill in the art would have found it obvious to modify the adaptive computing integrated circuit of New to reconfigure idle groups of computational elements, which includes generating reconfiguration information, in order to reduce the partial reconfiguration speed and increase overall throughput.

Conclusion

47. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

48. Wasson, U.S. Patent No. 6,433,578 has taught a heterogeneous programmable gate array with both structured and unstructured reconfigurable sub-arrays.

49. Lentz et al., U.S. Patent No. 6,272,579 has taught a microprocessor with multiple heterogeneous processors.

50. Sihlbom et al., U.S. Patent No. 6,653,859 has taught a heterogeneous digital signal processor with multiple reprogrammable logic cores.

51. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Barry J. O'Brien whose telephone number is (703) 305-5864. The examiner can normally be reached on Mon.-Fri. 6:30am-4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

52. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Barry J. O'Brien
Examiner
Art Unit 2183

BJO
7/12/2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100